



Wall

Jian-Jia construiește un perete din cărămizi de dimensiuni egale. Peretele este format din n coloane de cărămizi, numerotate de la 0 la $n - 1$ de la stânga la dreapta. Coloanele pot avea înălțimi diferite. Înălțimea unei coloane este numărul de cărămizi care o constituie.

Jian-Jia construiește peretele după cum urmează: inițial toate coloanele nu conțin nici o cărămidă. Apoi, Jian-Jia parcurge k faze *adăugând* sau *scăzând* cărămizi. Procesul de construire a peretelui se încheie atunci când toate cele k faze sunt parcurse. La fiecare fază lui Jian-Jia îi sunt date un interval de coloane consecutive și o înălțime h . El aplică apoi următoarea procedură:

- Într-o fază de *adăugare*, Jian-Jia adaugă cărămizi la acele coloane din intervalul dat care au mai puțin de h cărămizi, astfel încât să ajungă la înălțimea de exact h cărămizi. Coloanele care au h sau mai multe cărămizi rămân neschimbate.
- Într-o fază de *scădere*, Jian-Jia scoate cărămizi din acele coloane din intervalul dat care au mai mult de h cărămizi, astfel încât ele să ajungă la înălțimea de exact h cărămizi. Coloanele care au h sau mai puține cărămizi rămân neschimbate.

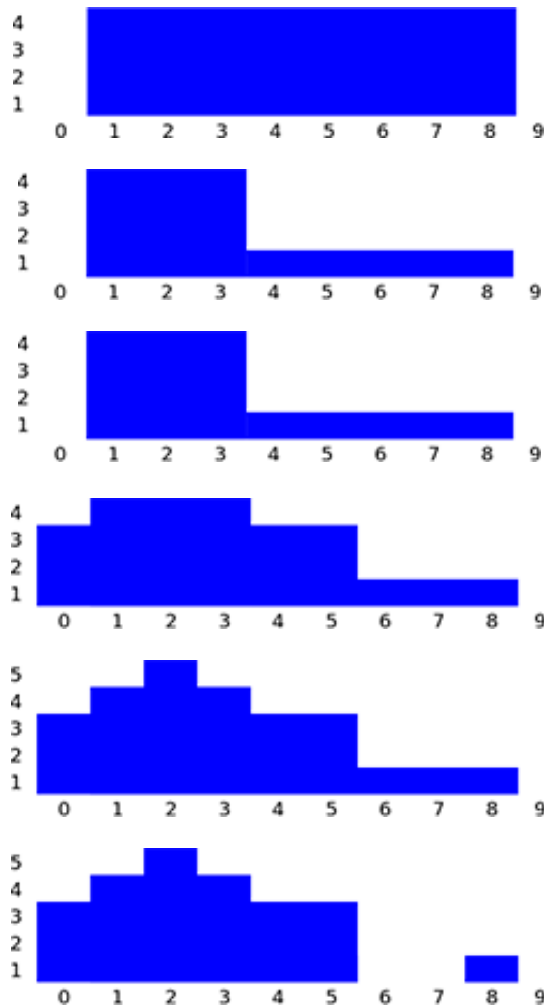
Sarcina voastră este să determinați forma finală a peretelui.

Exemplu

Să presupunem că avem 10 coloane din cărămizi și 6 faze de construire a peretelui. În tabelul de mai jos sunt incluse toate intervalele. Formele peretelui după fiecare fază sunt arătate mai jos.

faza	tipul	intervalul	înălțimea
0	adăugare	coloanele de la 1 la 8	4
1	scădere	coloanele de la 4 la 9	1
2	scădere	coloanele de la 3 la 6	5
3	adăugare	coloanele de la 0 la 5	3
4	adăugare	coloana 2	5
5	scădere	coloanele de la 6 la 7	0

După faza 0 fiecare coloană de la 1 la 8 va avea câte 4 cărămizi, deoarece toate coloanele inițial sunt goale. Coloanele 0 și 9 rămân în continuare goale. În faza 1, cărămizile sunt scoase de la coloanele de la 4 la 8 până când fiecare din coloane va avea câte o cărămidă, iar coloana 9 rămâne în continuare goală. Coloanele de la 0 la 3, care sunt în afara intervalului, rămân neschimbate. Faza 2 nu produce nici o schimbare, deoarece coloanele de la 3 la 6 nu au mai mult de 5 cărămizi. După faza 3, numărul de cărămizi în coloanele 0, 4, și 5 crește la 3. După faza 4, vom avea 5 cărămizi în coloana 2. Faza 5 elimină toate cărămizile din coloanele 6 și 7.



Cerință

Având descrierile celor k faze, vă rugăm să calculați numărul de cărămizi din fiecare coloană după parcurgerea tuturor fazelor. Trebuie să implementați funcția `buildWall`.

- `buildWall(n, k, op, left, right, height, finalHeight)`
 - n : numărul de coloane care formează peretele.
 - k : numărul de faze.
 - `op`: tablou unidimensional de lungime k ; `op[i]` este tipul fazei i : 1 pentru faza de adăugare și 2 pentru faza de scădere, pentru $0 \leq i \leq k - 1$.
 - `left` și `right`: tablouri unidimensionale de lungime k ; intervalul de coloane în faza i începe cu coloana `left[i]` și se termină cu coloana `right[i]` (inclusiv ambele capete `left[i]` și `right[i]`), pentru $0 \leq i \leq k - 1$. Se presupune că întodeauna `left[i] ≤ right[i]`.
 - `height`: tablou unidimensional de lungime k ; `height[i]` este parametrul de înălțime pentru faza i , pentru $0 \leq i \leq k - 1$.
 - `finalHeight`: tablou unidimensional de lungime n ; veți returna în acesta rezultatele obținute plasând numărul final de cărămizi din coloana i în `finalHeight[i]`, pentru $0 \leq i \leq n - 1$.

Subprobleme

Pentru toate subproblemele (subtask-urile) parametrii de înălțime din toate fazele sunt numere întregi nenegative mai mici sau egale cu **100,000**.

subproblemă	puncte	n	k	notă
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	nu există alte limitări
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	fazele de adăugare sunt parcurse înaintea fazelor de scădere
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	nu există alte limitări
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	nu există alte limitări

Detalii de implementare

Trebuie să încărcați exact un fișier, numit `wall.c`, `wall.cpp` sau `wall.pas`. În acest fișier se va implementa funcția descrisă mai sus, utilizând antetul de mai jos. De asemenea veți include un fișier header `wall.h` pentru programele C/C++.

pentru programele C/C++

```
void buildWall(int n, int k, int op[], int left[], int right[],  
int height[], int finalHeight[]);
```

pentru programele Pascal

```
procedure buildWall(n, k : longint; op, left, right, height :  
array of longint; var finalHeight : array of longint);
```

Grader-ul de pe computerul vostru

Grader-ul de pe computerul vostru citește datele de intrare în următorul format:

- linia 1: n, k .
- linia $2 + i$ ($0 \leq i \leq k - 1$): $op[i], left[i], right[i], height[i]$.